

Uncertainty aware anomaly detection to predict errant beam pulses in the Oak Ridge Spallation Neutron Source accelerator

Willem Blokland^{1,*}, Kishansingh Rajput², Malachi Schram², Torri Jeske², Pradeep Ramuhalli¹, Charles Peters¹, Yigit Yucesan¹ and Alexander Zhukov¹

¹Oak Ridge National Laboratory, Oak Ridge, Tennessee 37830, USA

²Thomas Jefferson National Accelerator Facility, Newport News, Virginia 23606, USA

 (Received 27 October 2021; accepted 7 November 2022; published 15 December 2022)

High-power particle accelerators are complex machines with thousands of pieces of equipment that are frequently running at the cutting edge of technology. In order to improve the day-to-day operations and maximize the delivery of the science, new analytical techniques are being explored for anomaly detection, classification, and prognostications. As such, we describe the application of an uncertainty aware Machine Learning method using the Siamese neural network model to predict upcoming errant beam pulses using the data from a single monitoring device. By predicting the upcoming failure, we can stop the accelerator before damage occurs. We describe the accelerator operation, related Machine Learning research, the prediction performance required to abort the beam while maintaining operations, the monitoring device and its data, and the uncertainty aware Siamese method and its results. These results show that the researched method can be applied to improve accelerator operations.

DOI: [10.1103/PhysRevAccelBeams.25.122802](https://doi.org/10.1103/PhysRevAccelBeams.25.122802)

I. INTRODUCTION

The Spallation Neutron Source (SNS) facility is the world's highest power proton accelerator, delivering 1.4 MW of a 1-GeV pulsed beam at 60 Hz. The beam is accelerated in the linear accelerator, which has both a warm, normal conducting and a cold, superconducting section. The accelerated beam is accumulated into the ring to form a very short but intense pulse with an intensity of up to 1.4×10^{14} protons. This short pulse is then sent to a stainless steel vessel filled with liquid mercury where the impact of the protons spalls the mercury atoms and neutrons are released [1]. The neutrons are then guided to experimental beam lines where the material research takes place.

Achieving high availability is extremely difficult in high-power proton beam accelerators. These accelerators use thousands of subsystems, with many running on the cutting edge of technology. Errant beam pulses can cause damage to the accelerator and negatively impact the research program [2]. To minimize downtimes, accelerator operations include preemptively replacing equipment, careful scheduling of maintenance periods, utilizing diagnostic

instruments to equipment, and detailed tracking of downtime statistics and patterning. These measures have limitations, as failures still happen unexpectedly. Adding more diagnostics instruments could help but is expensive.

Therefore, there is a need for methods that can utilize existing diagnostic data to identify the onset of errant beam pulses (see also [3–5] for using existing diagnostic data for adaptive controls tuning applied to particle accelerators). Errant beam pulses are caused by equipment failure as concluded by years of analysis of beam trip data. As equipment involved in the acceleration process must have an effect on the beam, we assume that conditions leading to errant beam pulses can be identified by monitoring signals from beam measurements.

This paper describes the results of research being conducted to exploit the extraordinary advantages of machine learning (ML) and the large amounts of accelerator data in a neutron production facility to improve accelerator availability. The focus of this research is on using ML to predict beam loss due to the failure of various accelerator equipment, using data (waveforms) from a single existing beam monitoring device. If successful, we can avoid the cost of installing additional monitoring devices to predict upcoming failures and, instead, use only existing diagnostics.

II. PREVIOUS WORK

Advances in computing power, the ability to stream and collect large data sets, and the availability of open source artificial intelligence initiatives have led to an increase in

*blokland@ornl.gov

Published by the American Physical Society under the terms of the *Creative Commons Attribution 4.0 International* license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

the use of ML methods in the accelerator and target community with the primary focus on improvements in beam tuning and beam quality [6,7]. Recent studies have evaluated the potential for improving accelerator operations by detecting deviations from normal or optimal conditions. Relevant literature focuses on building models of the beam behavior in order to identify scenarios that correspond to less than optimal accelerator setup or equipment errors. Fol *et al.* [8] provide an overview of several potential applications of ML for accelerators including optimization and prediction for tuning accelerator operations, lattice imperfection corrections, and anomaly detection. The paper discusses two specific applications of ML at the Large Hadron Collider (LHC): optics correction (predicting control knob setting to cancel quadrupole field errors) and anomaly detection [detecting faulty beam position monitors (BPMs)] [9,10]. Emma *et al.* [7] describe a multilayer perceptron-based prediction of the longitudinal phase space of particle accelerators based on diagnostic measurements, with good fault prediction accuracy. Similarly, [11] discusses a gradient boosting classifier for identifying beam loss plane contributions to the measured beam loss monitor (BLM) data at the LHC. Other studies discuss the application of ML for superconducting radio frequency (SRF) cavity fault classification [12], anomaly detection in superconducting magnets in the LHC [13], and for predicting shot-to-shot x-ray properties in an x-ray free-electron laser (XFEL) [14]. In all instances, the proposed ML appears to be capable of producing reasonable results, though there seems to be room for improvement in terms of true positive (TP) and false positive (FP) rates, particularly for an application such as ours that requires very low FP rates.

There are several studies using data from beam monitor sensors to identify conditions leading to beam loss. Rescic *et al.* [15,16] demonstrate that a random forest (RF) is capable of identifying a beam loss event at SNS one pulse in advance using data from one beam current monitor. In [17], the authors examine the potential to predict interlocks (beam interruptions) using multiple measurements along the accelerator and using a recurrence plots–based convolutional neural network (RPCNN) for transforming measurements into recurrence plots. They show that the performance of an RF and RPCNN is comparable, although the RPCNN is more successful at identifying anomalies that build up over time and in identifying the onset of faults earlier than the RF. The drawback is that the approach requires data from multiple measurements and may take additional computational time to generate the recurrence plots. The above discussion highlights the potential for using ML in accelerator applications for detecting anomalies.

This paper builds on the previous research in [15,16] by exploring a new machine learning technique and incorporating uncertainty quantification. The objective is to determine whether the beam pulse prior to the errant beam pulse is anomalous and can be detected. One of the challenges

associated with this problem is that a signal leading to an incoming faulty signal cannot be distinguished by visual inspection from a normal beam pulse signal and thus has very small differences. Another challenging aspect is the fact that in beam configuration changes may occur during the day-to-day operation of the accelerator, resulting in differences between two “normal” pulses. The range of possible configuration changes makes it difficult to compile a complete set of data from “normal” pulses. The resulting limitations in examples of normal data, and the small differences between normal and signal leading to an errant beam pulse are expected to challenge ML solutions (supervised and unsupervised) proposed to date for accelerator applications.

To overcome these challenges, this work applies a similarity-based supervised ML algorithm that compares reference known normal waveforms with an incoming waveform. Similar waveforms indicate that the incoming waveforms are normal. Additionally, the proposed model computes the uncertainty in its comparison to indicate incoming waveforms that have not been seen in the training data, which is critical for robust and reliable model predictions. Finally, in addition to the previously mentioned research in [15,16], a benchmark study is provided in the Appendix to compare the proposed model’s performance against classical autoencoder models for this problem. While autoencoders have shown promising results in the past, the method we are exploring exceeds these results as shown in the appendix.

III. DATA SET DESCRIPTION

A. Data source

The source of the data is the differential current monitor (DCM) [18,19], shown in Fig. 1. This system acquires beam current waveforms upstream and downstream of the superconducting cavity linac (SCL). Its main function is to protect the SCL from beam losses, which are detrimental to the superconducting cavities [20]. The system aborts the beam when it determines that less beam is coming out of the SCL than is going into the SCL. It does so faster than other devices, thanks to its dedicated communication line

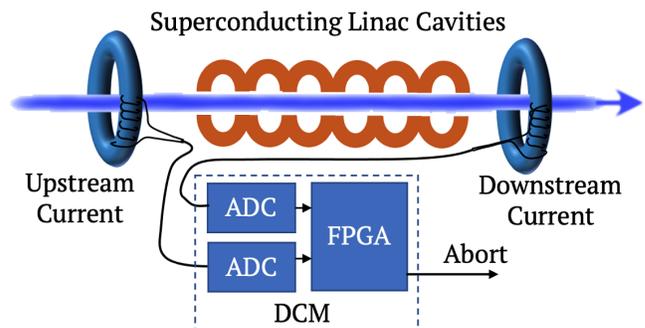


FIG. 1. Setup of the differential current monitor showing beam passing by the sensors and through the SCL.

with the front-end machine protection system (MPS) and the use of the field programmable gate array (FPGA). The FPGA also makes it very flexible to control how it aborts and when it archives. Data archived by the DCM are live streamed from the protected accelerator network to a remote accessible location for analysis.

B. Conditions for data archiving

The DCM archives the beam current waveforms not only when it notices the upstream/downstream difference but also if a beam current pulse is shorter than the previous beam current pulse. This means that it also archives when the beam is lost upstream or another device aborts the beam, therefore capturing practically all errant beam pulses. The DCM will not archive if the accelerator is aborted between beam pulses. While a single shorter beam pulse does not directly cause SCL beam losses, the RF cavity controllers will learn the wrong beam loading for the next normal length pulse which will then result in beam loss in the SCL, see [21,22]. The DCM prevents this upcoming beam loss by alerting the MPS (because the beam pulse was shorter), which then results in the RF controller reusing the previous full length beam pulse for its feed-forward setup. We will refer to the anomalous pulse without beam loss as a 0011 event and the anomalous beam pulse with SCL beam loss as an 1111 event.

C. Data set contents

A unique feature of the DCM, one that makes it possible to use it for ML, is that it not only archives the beam current waveforms of the aborted beam pulses in the data set but also the beam current waveforms before the errant beam pulse and additionally regularly archives nonerrant beam current waveforms. The before waveforms can be used for (semi)-supervised learning as the anomalous class, while the nonerrant beam current waveforms can be used as the normal class.

An example of the beam current waveform is shown in Fig. 2. The upper plot shows what is referred to as a series of macropulses, a 1-ms long pulse repeated at 60 Hz. This macropulse consists of approximately 1000 minipulses. Each minipulse is ~ 650 ns and is followed by a gap of ~ 350 ns. Within each minipulse are the micropulses, not shown, which are the RF buckets filled with the beam particles spaced at 402.5 MHz. The bottom plot shows a beam pulse's current waveform with the initial ramp-up in intensity in the beginning of the macropulse, as well as the different widths of the minipulses during the macropulse. This setup is typical for the production of style beam.

The waveform has a length of about 1.2 ms and is digitized at 100 MS/s. Along with the upstream and downstream beam current waveforms, the beam pulse's timestamp and cycle ID are collected and archived.

The data set for the ML training and testing was comprised of 4000 normal beam pulses and 4000

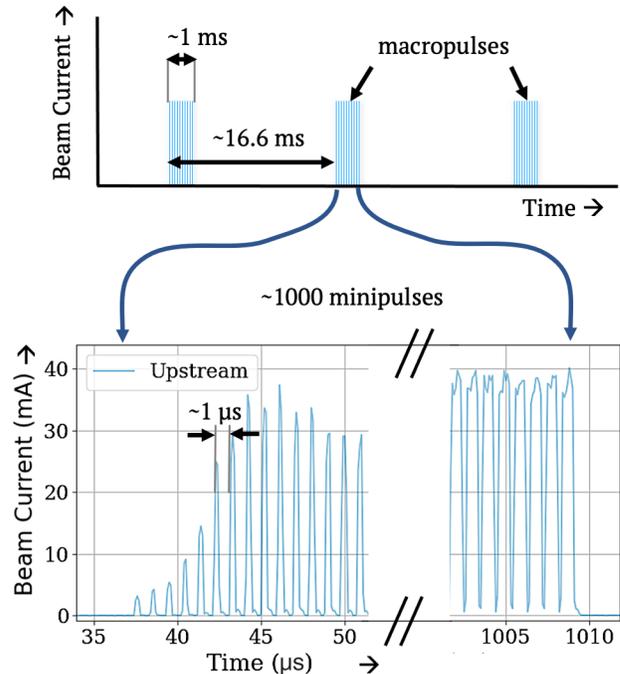


FIG. 2. Beam macropulse pattern and beam current waveform.

anomalous pulses. We excluded the anomalous pulses with beam loss from the training set but included these in an extended test set to be able to determine if this type of anomalous beam pulse can also be detected with the Siamese model but with higher uncertainty.

IV. METHODS

A. Siamese model

For this study, we explore the use of Siamese Neural Network (SNN) models [23] to provide a natural similarity ranking between two inputs. The SNN learns the similarities between two input waveforms in contrast to a traditional classification approach that learns to classify each data type. The SNN model developed in this paper includes a final layer to approximate a Gaussian process (GP) model in order to provide an out-of-distribution uncertainty estimate. The SNN model with GP provides model flexibility to include multiple input data modalities originating from different sensors (beam position monitors, beam phase monitors, etc). The SNN approach allows for a limited amount of monitoring devices to identify beam loss events, avoids extensive computational loads, can adjust the threshold to change the TP and FP rates, can determine if the trained model is outdated, and can detect anomalies that are not in the training set from normal beam pulses. Our aim is to use the Siamese model to learn the similarities between normal and anomalous beam pulses, as measured by the DCM beam current sensors. Developing an ML model based on a similarity score provides robustness against previously unseen anomalies that could be

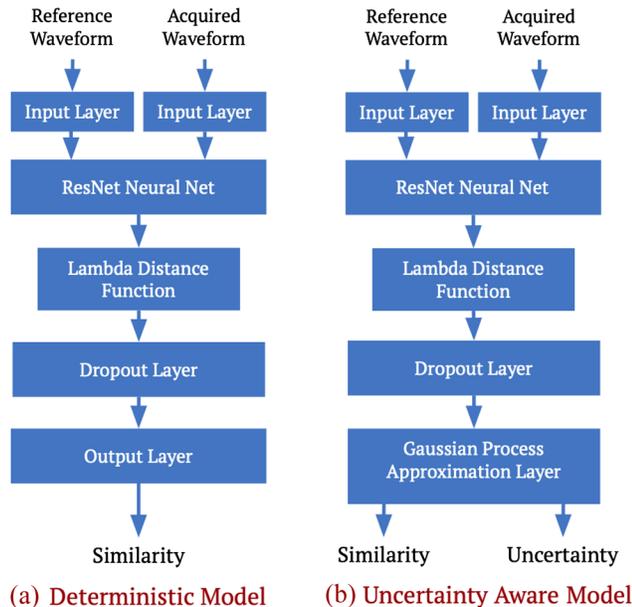


FIG. 3. The deterministic Siamese model architecture (a) is composed of two input layers for the reference waveforms and the acquired waveform, followed by one common ResNet model and a distance function. Then a dropout layer is applied to avoid overfitting, and finally an output layer to compress the dimensionality and produce the similarity scalar. The uncertainty aware model (b) replaces the output layer with a Gaussian process approximation layer in order to produce both the similarity and the uncertainty values.

introduced to the system. The similarity score can be used to reevaluate the applicability of the current model by comparing a normal beam pulse used for training with the current normal pulse. A trending change in the similarity score would indicate the need to retrain the current Siamese model. Alternatively, using an updated normal waveform, not from the training set, for comparison with the input waveform can possibly extend the life of the trained model.

A Siamese model consists of twin networks that accept unique inputs, as illustrated in (A) of Fig. 3. The twin networks are used to shrink the large amount of raw data to a reduced representation that captures the salient features. The reduced representations of each input are then compared using a modified contrastive loss function [24]:

$$L(y, \hat{y}) = \alpha \times (1 - y) * \hat{y}^2 + y * [\max(\beta - \hat{y}, 0)]^2 \quad (1)$$

The contrastive loss function is composed of two terms used to decrease the output of like pairs and increase the output of unlike pairs. Here y is the truth value, \hat{y} is the predicted value, α is tuning parameter use to emphasize the similar pulses that were set to 2 for this study, and β is a second tuning parameter, set to 1, used to emphasize dissimilar pulse. We used a ResNets [25] model for the twin network. ResNets consist of several stacked residual units, which can be thought of as a collection of convolutional layers coupled with a shortcut that improves the propagation of the signal in

a neural network. This shortcut allows for the construction of much deeper networks since keeping a clean information path in the network facilitates optimization.

The model was developed using KERAS [26] and TensorFlow back end [27]. We used the Adam optimizer [28] and a loss function defined in Eq. (1). For our study, we used the similarity metric as defined in Eq. (2):

$$L^2 = \left| \sum_{i=0}^N (x_{1,i}^2 - x_{2,i}^2) \right| \quad (2)$$

Here x_1 and x_2 are the latent vector outputs from the ResNet model for input pulse 1 and 2, and i is the element index.

B. Uncertainty aware Siamese model

Providing methods to reliably quantify the predictive uncertainty for our models is critical for real-world applications. This is acutely visible when the input samples are dissimilar to the training sample. The use of distance awareness is particularly important because deterministic models are only trained on a data set $\mathcal{D} = \{y_i, \mathbf{x}_i\}_{i=1}^N$ where \mathcal{D} is a subset of the input space, $\mathcal{H}_{IDD} \subset \mathcal{H}$. Consequently, the model only learns the in-domain distributions, $p^*(y|\mathbf{x}, \mathbf{x} \in \mathcal{H}_{IDD})$ from the data set \mathcal{D} , and there is a possibility of an orthogonal data set, $p^*(y|\mathbf{x}, \mathbf{x} \notin \mathcal{H}_{IDD})$, existing in the same input space. This orthogonal distribution $p^*(y|\mathbf{x}, \mathbf{x} \notin \mathcal{H}_{IDD})$ can be very different from the training distribution, and the predictions on this orthogonal set can lead to unreliable results.

For this study, we extend the Siamese model described in Sec. IV A by replacing the output layer with an approximation to Gaussian process (GP) as described in [29] but applied to the Siamese model. A classic deep learning model maps the input space to a hidden representation space and its output layer maps the hidden representation $h(x)$ to the label space y . By wrapping a GP layer around the output layer, we make it distance-aware such that it outputs an uncertainty score representing the distance between the hidden space of the test data to that of the input space (distribution that the model is trained on) $|h(x) - h(x')|$. The updated architecture is shown in (b) of Fig. 3.

V. ML PERFORMANCE METRICS

We must determine the ML performance metrics in terms of FP and TP rates required to maintain and improve accelerator performance. One of the key metrics to track the performance of the SNS facility is beam availability. To prevent a negative impact on scientific research, any method that can wrongly abort the accelerator beam should not noticeably increase the current levels of beam trip frequency or duration. While many ML applications often allow single digit percentages of FP rates, in the SNS case, a 5% wrongly aborted beam would reduce our integrated power and adversely affect the material science research

program. We will define an acceptable FP rate by analyzing the number and duration of recent beam trips using data from the DCM.

If there is beam loss in the SCL, then the beam is held off until an operator can verify it is safe to start up again. For the month of March 2021, the total beam production time was ≈ 26.4 days with an average daily trip frequency of SCL beam loss trips of ≈ 5 per day, a total of 126 trips. When including the recovery time to ramp-up to full beam power, about 0.22% of beam pulses were lost. There were many more events of truncated beam pulses, about 5500 trips, for a total downtime of about 1.3% over the data set. In addition, each type of event can lead to much longer downtimes on the order of hours. While preventing these errant beam pulses might prevent these long downtimes as well, we will not include these downtimes in our ML performance criteria because of their rare occurrence. We assume that a missed beam rate of 1.5% (0.22% + 1.3%) is an acceptable level and verified with operations that an increase of up to 20% is acceptable and well within month-to-month variations. The ML technique can then have an FP level that contributes to around 0.2% of beam pulses missed. A TP rate of 0.5 in case of the beam loss events will gain us 0.1% of the beam. Any TP rate for truncated errant beam pulses will not lower SCL losses as the loss of beam is upstream. Preventing a truncated beam pulse will have as benefit that the accelerator will not send an irregular lower intensity pulse to the target. After a truncated pulse, the accelerator is automatically re-enabled after four missed beam pulses. If we predict a truncated pulse, we would still get the same 4 pulse hold-off period unless the MPS is modified to reduce the hold-off time for predicted errant beam pulses. We can increase the beam to target only if the hold off is reduced. The current minimal hold off of the beam is 4 pulses thus the FP rate of truncated beam pulses should be $0.2\%/4$ or 0.05%. This would put our recall, $TP/(TP + FN)$, at 0.5 and our precision, $TP/(TP + FP)$, at 0.999.

VI. RESULTS

For this study, we used the receiver operating characteristic (ROC) curve to quantify the performance of our models. The ROC curve indicates the relationship between true positives (TP) and false positives (FP). In our case, a true positive is defined as correctly identifying an anomaly and a false positive is defined as incorrectly identifying a normal pulse as anomalous. We trained a deterministic and a distance-aware Siamese model with identical architectures except for the output layer as explained in Sec. IV.

A. Data preprocessing

Example waveforms are shown in Fig. 4. The before waveform (blue) is used for training. The errant beam pulse (orange) shows a drop in beam current before it is aborted.

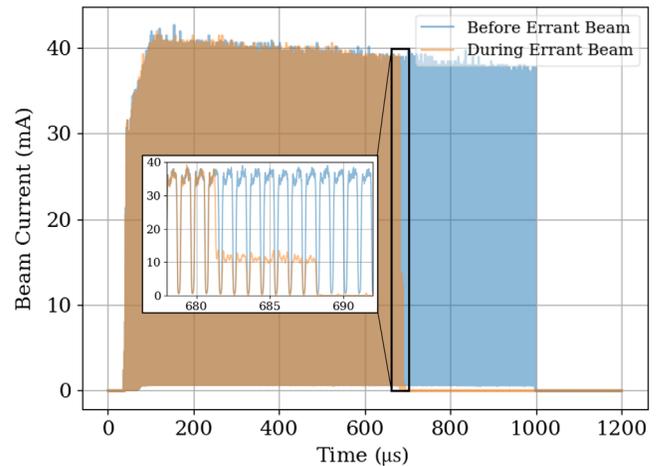


FIG. 4. Digitized waveform of the beam current before the errant beam pulse and the errant beam pulse itself.

Each waveform contains 120,000 elements, of which elements 3000–13,000 are used for the Siamese model as we determined that this section holds the information on whether a pulse is normal or anomalous.

Figure 5 displays the box plots for 20 randomly selected waveforms from the normal set of waveforms. To remove waveforms from the data set that are not part of the normal neutron production, we apply preprocessing based on the number of minipulses in the waveform and the beam repetition rate. Peaks were identified using the `find_peaks` method from the SciPy library [30], with a minimum height of 2 mA and a minimum distance of 75 elements (750 ns) between two neighboring peaks. An example waveform demonstrating the identified peaks is shown in Fig. 6. In addition, we require the macropulse to have at least 900 minipulses and to be repeated at 60 Hz.

To generate the data set for the Siamese model, we extracted normal and anomalous “Before” waveforms from

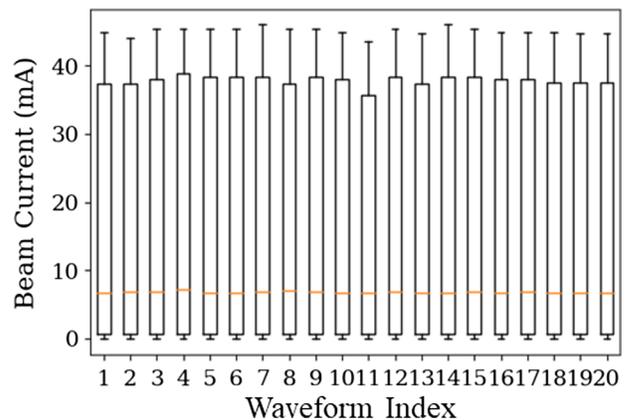


FIG. 5. Box plot displaying a random sample of 20 waveforms from the training set. The orange horizontal line corresponds to the median, the upper (lower) edge of the whisker corresponds to the maximum (minimum) value, and the upper (lower) edge of the box denotes the third (first) quartiles.

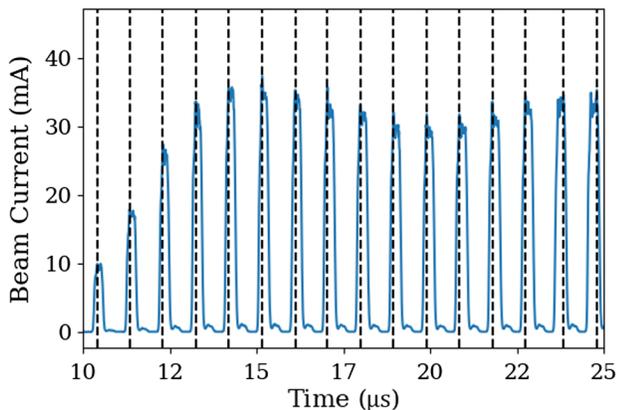


FIG. 6. Example results using `scipy.signal.find_peaks` function with a minimum height of 2 mA and minimum distance of 75 elements between neighboring peaks.

the archived data from March 2021 from the upstream sensor. From this, after preprocessing to exclude non-production beam pulses, we selected 4000 anomalous waveforms and compared each of them to 15 randomly selected normal waveforms. We used 4000 normal waveforms and also compared each to 15 randomly selected normal waveforms. The comparisons between normal waveforms are assigned a label of 0 and comparisons between normal and anomalous waveforms are assigned a label of 1. The data are then divided into orthogonal training, testing, and validation data sets that contain 76,800, 24,000, and 19,200 samples, respectively. Here, a sample is defined as a combination of normal and/or anomalous waveforms used as an input to the Siamese model. The training, testing, and validation data sets contain equal numbers of normal-to-normal and normal-to-anomalous samples.

B. Deterministic Siamese model results

Figure 7 displays the results of the classification. The model identifies most of the anomalies with very high confidence, while the remaining anomalies are misclassified as normal.

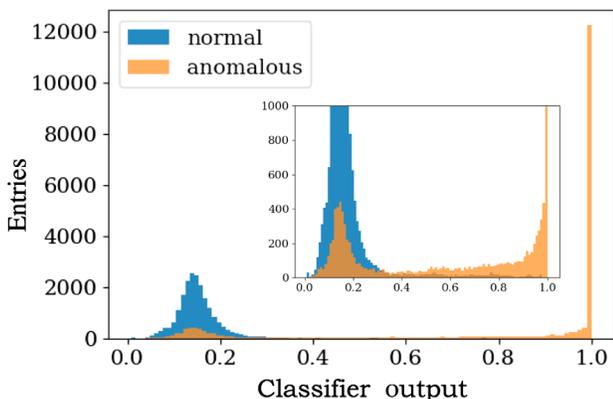


FIG. 7. Classifier output histogram (the subplot represents a zoomed-in version of the same histogram plot).

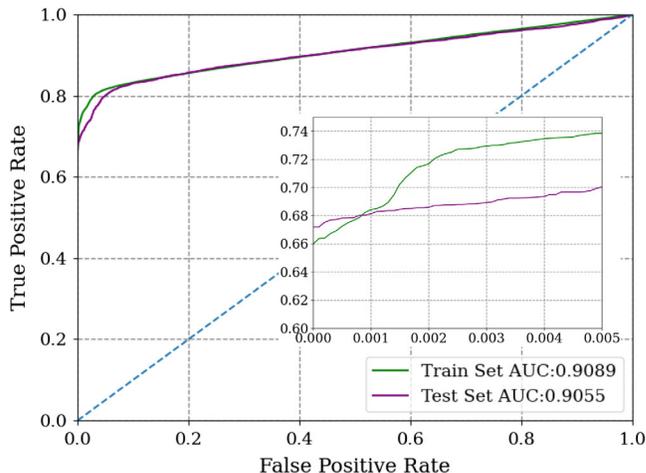


FIG. 8. Deterministic model ROC curves for train and test samples.

For the developed solution to be of practical use, we must identify the maximum number of correctly identified anomalies while maintaining the FP rate below the established 0.05%. As displayed in Fig. 8 (zoomed-in subplot), we have a true positive rate of more than 60% on both train and test data sets.

C. Uncertainty aware Siamese model results

To implement the uncertainties associated with the outputs of our Siamese model, we wrapped the last layer of the model with a Gaussian process layer. The uncertainty aware Siamese model not only provides a classifier but also outputs the uncertainty of the predictions. We can explore how the model behaves in both dimensions.

As shown in Fig. 9, we introduced beam pulses (red dots) with anomalies that the model was not used on during the

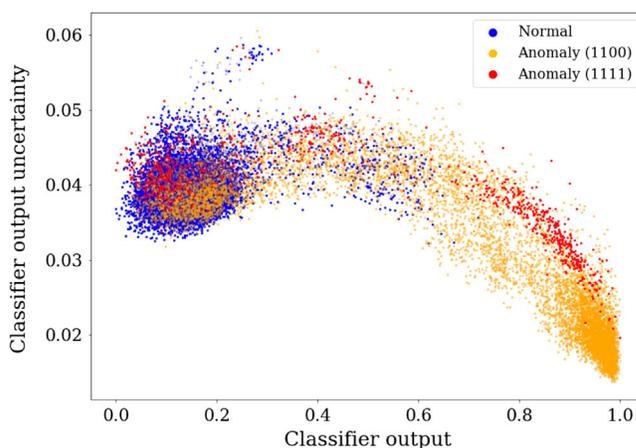


FIG. 9. Model predicted uncertainty versus uncertainty aware model prediction on a test data set. The blue dots are the normal pulses, the orange dots are the anomaly type used for training, and the red dots are the anomalous pulses that the model was not trained on.

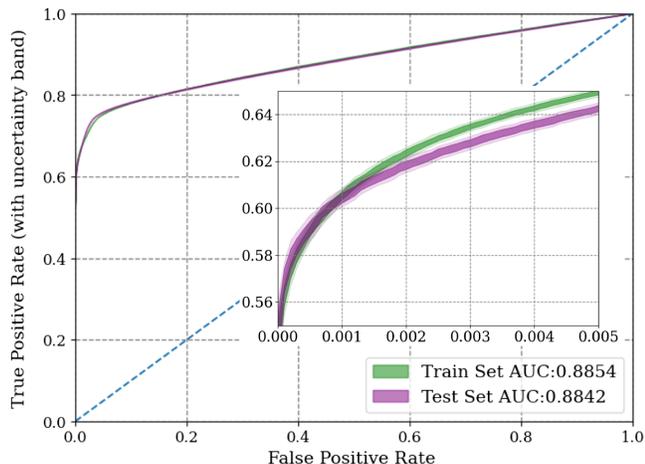


FIG. 10. Uncertainty aware model ROC curves for train and test samples. Dark error bands are for the 25%/75% quantiles and the light color bands are for the 10%/90% values.

model training. The model performs comparably in terms of TP and FP to the originally trained anomalies, however, with larger uncertainties. We interpret the increase in the uncertainty as due to the difference in the samples corresponding to these anomalies from the training samples, thereby increasing the out-of-distribution uncertainty.

In order to incorporate the model prediction uncertainties into the ROC curve, we smeared the model prediction output with its associated uncertainty using a Gaussian distribution. We conducted 250 trials to compute the ROC curve bands, as shown in Fig. 10. The dark bands represent the range between the 25th and 75th quantiles and the lighter bands represent the range between the 10th and 90th quantiles.

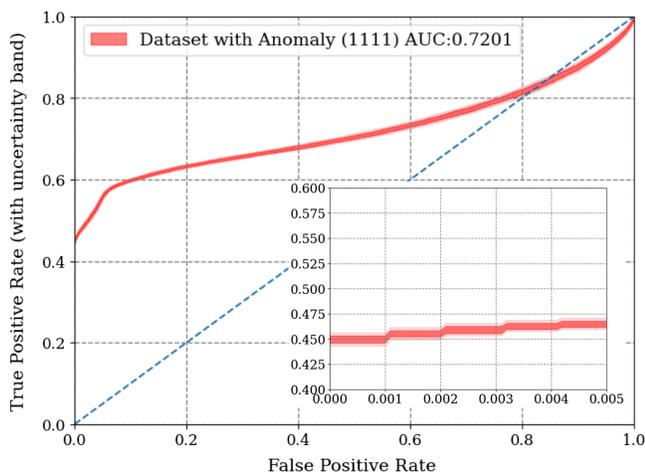


FIG. 11. ROC curve with uncertainty band for the inferences made on anomaly type 1111 with the model trained on anomaly type 1100. The dark and light bands represent the same range as Fig. 10.

TABLE I. Expected model inference time for select computing hardware. Core i9-9880H and Core i5-4400E times were measured.

Intel CPU	CPU mark rating	Inference time (ms)	
		Deterministic	Uncertainty
Core i5-4400E	3251	3.2/NA	(C++)
Core i7-5700	5905	4.8/9.5	
Core i9-9880H	14,075	2.0/4.0	
Xeon E5-2618	10,464	2.7/5.4	
Xeon W-2245	19,527	1.4/2.9	

Figure 11 shows the ROC curve uncertainty band for the unseen anomalies (type 1111 aka beam loss events) discussed above. Even though the model was not trained on these anomalies, it is able to identify more than 45% of the anomalies correctly while keeping the false positive below the threshold of 0.05%, although the predictions have higher uncertainties as seen in Fig. 9. The uncertainty bands in Figs. 10 and 11 provide a quantifiable estimation of the model's ability to identify existing and/or new anomalies.

It should be noted that the scatterplot also shows that there is a threshold where the FP rate is negligibly small.

D. Siamese inference CPU timing results

The Siamese model has been tested on an Intel Core i9, with the timing results coming out at about an average of 2 ms per inference for the deterministic model and 4 ms for the distance-aware model. The current DCM CPU is a Core i7 CPU and based on [31], it would still be able to infer within 10 ms. It should be noted that the DCM is scheduled to be upgraded to one of the Xeon CPU (W-2245 or E5-2618) listed in Table I. As such, all CPUs considered are expected to easily complete the inference within the allotted time. As the data are transferred by DMA over a PXIe bus on a point-by-point basis, during sampling, the data are almost instantly available to the real-time CPU for processing. Current CPU usage is around 2–3 ms per 16.6 ms cycle. That means 16.6–1–3, or 12.6 ms, of CPU time, is available to convert the data from fixed point to float, evaluate with the Siamese model on the real-time OS on the CPU, and send an abort signal back to the FPGA to abort the accelerator. We also tested the deterministic model inference running as a C++ code on NI PXIe-8840 (Core i5-4400E CPU) system with LabVIEW Real-Time OS, one inference took under 4 ms.

VII. DISCUSSION

A. Siamese improvements

In addition to the uncertainty quantification, we aim to implement a class activation map (CAM) to highlight the distinct region(s) of the pulse the model focuses on when making a similarity classification. This method has been used extensively in recent years upon the realization that

convolutional neural networks can perform object localization without explicit supervision of the object [32]. This can be used to determine specific equipment failure classes that can be compared to failed equipment as indicated by the MPS.

B. Implementation

Given the performance of the method in both execution times and TP and FP rates, we plan to implement the model on the actual DCM system in its real-time system. In this paper, we only analyzed the beam pulse immediately before the fault. However, the DCM also archives up to 25 preceding pulses. As such, we plan to study if the preceding pulses can also provide additional discriminating power which would allow us to detect a trend and thus identify faults earlier and more accurately. We plan to determine whether different equipment failures have a different duration from the first detection of an anomaly to the actual errant beam pulse and then apply the appropriate hold-off time for each different type of equipment failure. If the anomaly can be detected early enough, alternative methods, such as adjusting the equipment that is about to fail, can be used instead of aborting the beam. We are also investigating to see if we can implement the Siamese model inference on the FPGA to provide fast results and run more inferences of the incoming waveform versus multiple reference waveforms.

We now also have data becoming available from the beam position monitors (BPMs). The phase data from the BPMs are especially interesting, as the phase relates directly to the momentum of the beam particles and the momentum is directly related to the acceleration process. Thus if the acceleration process is failing, we should see this in the phase data immediately. We hope that in this data even more precursors can be found.

VIII. CONCLUSION

In this paper, we developed and applied an uncertainty aware method to predict impending faults using data from a single data source. Using the unique uncertainty aware Siamese model, we can set the FP rate low enough, 0.05%, so that the performance of the accelerator is insignificantly affected while maintaining a TP rate high enough, 60% for errant beam events in the training set and 45% for events not in the training set, to benefit the accelerator. This result exceeds previously published studies, [15] and [16], by reducing the FP rates while improving the TP. Another feature of the Siamese model is that we can use it to determine whether the latest normal pulse is still similar to the trained normal reference pulse. This comparison can help us determine whether we need to retrain the model. Execution times of the model are such that practical implementation is possible, which will help us determine the benefits of preventing errant beam.

ACKNOWLEDGMENTS

The authors acknowledge the help from David Brown in evaluating Operations requirements, Frank Liu, for his assistance on the ML techniques, and Sarah Cousineau for making this grant work possible. The authors are grateful for support from the Neutron Sciences Directorate at ORNL in the investigation of this work. This work was supported by the DOE Office of Science, United States under Grant No. DE-SC0009915 (Office of Basic Energy Sciences, Scientific User Facilities program). This research used resources at the Spallation Neutron Source, a DOE Office of Science User Facility operated by the Oak Ridge National Laboratory. This manuscript has been authored by UT-Battelle, LLC, under Contract No. DE-AC05-00OR22725 and the Jefferson Science Associates (JSA) under Contract No. DE-AC05-06OR2317 with the US Department of Energy (DOE). The US government retains and the publisher, by accepting the article for publication, acknowledges that the US government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for US government purposes. DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan [33].

APPENDIX: COMPARISON BETWEEN SIAMESE MODEL AND CONVOLUTIONAL AUTOENCODERS

Autoencoders are widely adopted ML techniques utilized for fault detection and recently applied for particle accelerators [34–36]. This section provides results of a comparative analysis of proposed Siamese Neural Network (SNN) against Convolutional AutoEncoders (CAE).

In this analysis, we trained nine CAE architectures with varying depth and parameters (see Table II for architecture details) using the same data set used to train the Siamese

TABLE II. Network details for autoencoder architectures used in comparative study. Note that for the number of layers and filters, we only report the encoder part, as the decoder is symmetric to the encoder.

Arc. No.	No. of Layers	Filters	Kernel size	Activation function	Parameters
1	4	256/128/64/32	8	ReLU	702,721
2	3	128/64/32	8	ReLU	175,105
3	2	64/32	8	ReLU	42,309
4	4	256/128/64/32	12	ReLU	1,052,929
5	3	128/64/32	12	ReLU	262,145
6	2	64/32	12	ReLU	63,361
7	4	256/128/64/32	10	Sigmoid	877,825
8	3	128/64/32	10	Sigmoid	218,625
9	2	64/32	10	Sigmoid	52,865

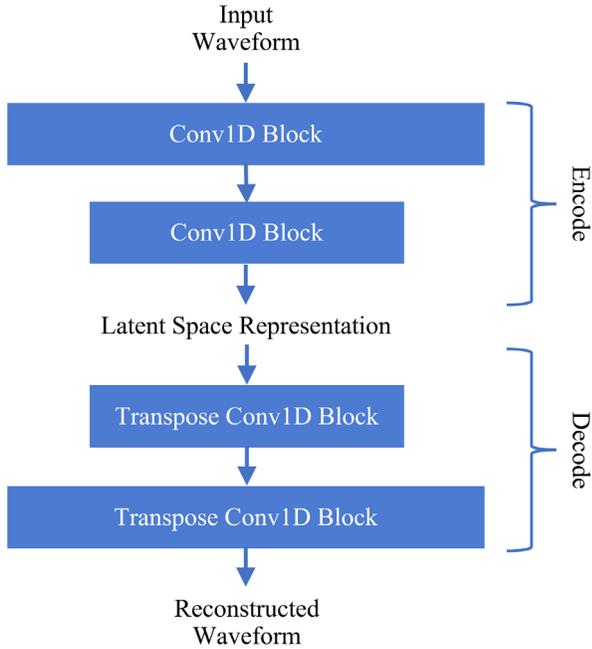


FIG. 12. Block diagram of a generic Convolutional AutoEncoder architecture.

Twin model in this work. Figure 12 illustrates a generic autoencoder model architecture, where each convolutional block includes sequential layers of Conv1D (transpose for the decoder stage), batch normalization, and dropout. We tested model architectures that have an equivalent or greater number of parameters as was used for SNN in this paper. CAEs are trained with normal waveforms in order to reconstruct the signal. Trained models are then used to predict the reconstruction of both normal and anomalous waveforms of the test set. Predicted waveforms are

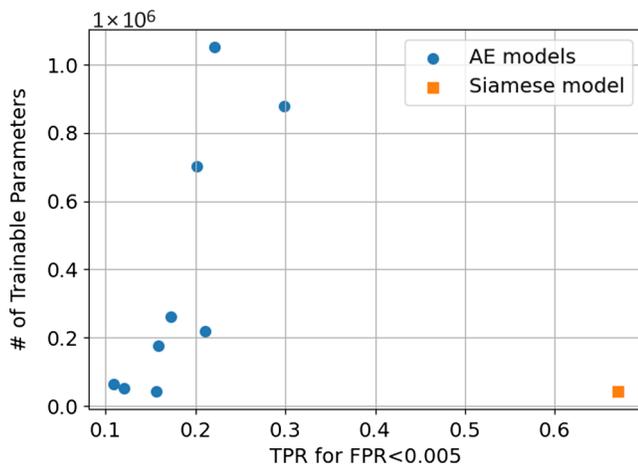


FIG. 13. Performance comparison of multiple Convolutional Autoencoder architectures against the proposed SNN. The horizontal axis is the maximum TPR below the predetermined FPR threshold, while the vertical axis represents model size in terms of the number of trainable parameters.

compared against input waveforms and a reconstruction error is obtained using root mean squared error [as formulated in Eq. (A1)].

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=0}^N (x_i - \hat{x}_i)^2}, \quad (\text{A1})$$

where x is the input waveform vector, \hat{x} is the reconstructed waveform vector, and N is the length of the waveform. This metric is used to build ROC curves and the maximum TP rate (TPR) for the target FP rate (FPR) is extracted.

Figure 13 depicts the comparison of CAE designs against the SNN in terms of model size (No. of trainable parameters within the model) and predictive performance. This analysis shows that it is possible to improve the performance of CAE by simply increasing the complexity of the model. However, even the most complex CAE design used in this analysis is outperformed more than twice by the SNN, which consists of multiple orders of magnitude less parameters.

- [1] S. Henderson *et al.*, The spallation neutron source accelerator system design, *Nucl. Instrum. Methods Phys. Res., Sect. A* **763**, 610 (2014).
- [2] C. Peters, W. Blokland, T. Justice, and T. Southern, Minimizing errant beam at the Spallation Neutron Source, in *Proceedings of 6th International Beam Instrumentation Conference, Grand Rapids, MI (JACoW, Geneva, Switzerland, 2018)*, TH2AB1.
- [3] A. Scheinker, E.-C. Huang, and C. E. Taylor, Extremum seeking-based control system for particle accelerator beam loss minimization, *IEEE Trans. Control Syst. Technol.* **30**, 2261 (2021).
- [4] A. Scheinker, A. Edelen, D. Bohler, C. Emma, and A. Lutman, Demonstration of Model-Independent Control of the Longitudinal Phase Space of Electron Beams in the Linac-Coherent Light Source with Femtosecond Resolution, *Phys. Rev. Lett.* **121**, 044801 (2018).
- [5] A. Scheinker and S. Gessner, Adaptive method for electron bunch profile prediction, *Phys. Rev. ST Accel. Beams* **18**, 102801 (2015).
- [6] S. C. Leemann, S. Liu, A. Hexemer, M. A. Marcus, C. N. Melton, H. Nishimura, and C. Sun, Demonstration of Machine Learning-Based Model-Independent Stabilization of Source Properties in Synchrotron Light Sources, *Phys. Rev. Lett.* **123**, 194801 (2019).
- [7] C. Emma, A. Edelen, M. J. Hogan, B. O'Shea, G. White, and V. Yakimenko, Machine learning-based longitudinal phase space prediction of particle accelerators, *Phys. Rev. Accel. Beams* **21**, 112802 (2018).
- [8] E. Fol, R. Tomás, G. Franchetti, and J. M. Coello de Portugal, Application of machine learning to beam diagnostics, in *Proceedings of 39th Free Electron Laser Conference, FEL2019, Hamburg, Germany (JACoW, Geneva, Switzerland, 2018)*, pp. 169–176.

- [9] E. Fol, J. M. Coello de Portugal, and R. Tomás, Unsupervised machine learning for detection of faulty beam position monitors, in *Proceedings of 10th International Particle Accelerator Conference, Melbourne, Australia, IPAC2019* (JACoW, Geneva, Switzerland, 2019), pp. 2668–2671.
- [10] E. Fol, R. Tomás, J. Coello de Portugal, and G. Franchetti, Detection of faulty beam position monitors using unsupervised learning, *Phys. Rev. Accel. Beams* **23**, 102805 (2020).
- [11] G. Valentino and B. Salvachua, Machine learning applied at the LHC for beam loss pattern classification, *J. Phys. Conf. Ser.* **1067**, 072036 (2018).
- [12] C. Tennant, A. Carpenter, T. Powers, A. Shabalina Solopova, L. Vidyaratne, and K. Iftekharuddin, Superconducting radio-frequency cavity fault classification using machine learning at Jefferson laboratory, *Phys. Rev. Accel. Beams* **23**, 114601 (2020).
- [13] M. Wielgosz and A. Skoczzeń, Using neural networks with data quantization for time series analysis in LHC superconducting magnets, *Int. J. Appl. Math. Comput. Sci.* **29**, 503 (2019).
- [14] A. Sanchez-Gonzalez *et al.*, Accurate prediction of x-ray pulse properties from a free-electron laser using machine learning, *Nat. Commun.* **8**, 15461 (2017).
- [15] M. Rei, R. Seviour, and W. Blokland, Predicting particle accelerator failures using binary classifiers, *Nucl. Instrum. Methods Phys. Res., Sect. A* **955**, 163240 (2020).
- [16] M. Rei, R. Seviour, and W. Blokland, Improvements of preemptive identification of particle accelerator failures using binary classifiers and dimensionality reduction, *Nucl. Instrum. Methods Phys. Res., Sect. A* **1025**, 166064 (2022).
- [17] S. Li, M. Zacharias, J. Snuverink, J. Coello de Portugal, F. Perez-Cruz, D. Reggiani, and A. Adelmann, A novel approach for classification and forecasting of time series in particle accelerators, *Information* **12**, 121 (2021).
- [18] W. Blokland and C. C. Peters, A new differential and errant beam current monitor for the SNS* accelerator, in *Proceedings of 2nd International Beam Instrumentation Conference, IBIC'13, Oxford, United Kingdom* (JACoW, Geneva, Switzerland, 2013), pp. 921–924.
- [19] W. Blokland, C. Peters, and T. Southern, Enhancements to the SNS* differential current monitor to minimize errant beam, in *Proceedings of 8th International Beam Instrumentation Conference, IBIC'19, Malmö, Sweden* (JACoW, Geneva, Switzerland, 2019), pp. 146–149, [10.18429/JACoW-IBIC2019-MOPP025](https://doi.org/10.18429/JACoW-IBIC2019-MOPP025).
- [20] S.-H. Kim *et al.*, The status of the superconducting linac and SRF activities at the SNS, in *Proceedings of the 16th International Conference on RF Superconductivity, SRF2013, Paris, France* (JACoW, Geneva, Switzerland, 2014), pp. 83–88.
- [21] S.-H. Kim, R. Afanador, D. Barnhart, M. Crofford, B. Degraff, M. Doleans, J. Galambos, S. Gold, M. Howell, J. Mammosser, C. McMahan, T. Neustadt, C. Peters, J. Saunders, W. Strong, D. Vandygriff, and D. Vandygriff, Overview of ten-year operation of the superconducting linear accelerator at the spallation neutron source, *Nucl. Instrum. Methods Phys. Res., Sect. A* **852**, 20 (2017).
- [22] H. Ma, M. Champion, M. Crofford, K. Kasemir, M. Piller, L. Doolittle, and A. Ratti, Low-level rf control of spallation neutron source: System and characterization, *Phys. Rev. ST Accel. Beams* **9**, 032001 (2006).
- [23] G. Koch, R. Zemel, R. Salakhutdinov *et al.*, Siamese neural networks for one-shot image recognition, in *Proceedings of ICML Deep Learning Workshop, Lille* (2015), Vol. 2, <https://www.cs.cmu.edu/~rsalakhu/papers/oneshot1.pdf>.
- [24] R. Hadsell, S. Chopra, and Y. LeCun, Dimensionality reduction by learning an invariant mapping, in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR'06, New York, NY* (IEEE, New York, 2006), Vol. 2, pp. 1735–1742.
- [25] K. He, X. Zhang, S. Ren, and J. Sun, Deep residual learning for image recognition, in *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV* (IEEE, New York, 2016), p. 770.
- [26] F. Chollet *et al.*, Keras, <https://keras.io> (2015).
- [27] M. Abadi *et al.*, TensorFlow: Large-scale machine learning on heterogeneous systems (2015), software available from tensorflow.org.
- [28] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, in *Proceedings of 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA* (2015), [arXiv:1412.6980](https://arxiv.org/abs/1412.6980).
- [29] J. Z. Liu, Z. Lin, S. Padhy, D. Tran, T. Bedrax-Weiss, and B. Lakshminarayanan, Simple and principled uncertainty estimation with deterministic deep learning via distance awareness, [arXiv:2006.10108](https://arxiv.org/abs/2006.10108).
- [30] P. Virtanen, R. Gommers, T. E. Oliphant *et al.*, SciPy 1.0: fundamental algorithms for scientific computing in Python, *Nat. Methods* **17**, 261 (2020).
- [31] CPU Benchmark, <https://www.cpubenchmark.net/compare/> (2021) [accessed October 5, 2021].
- [32] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, Learning deep features for discriminative localization, in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV* (IEEE, New York, 2016), pp. 2921–2929.
- [33] https://www.energy.gov/sites/prod/files/2014/08/f18/DOE_Public_Access%20Plan_FINAL.pdf.
- [34] J. P. Edelen and N. M. Cook, Anomaly detection in particle accelerators using autoencoders, in *Proceedings of the 2021 Improving Scientific Software Conference* (No. NCAR/TN-567+PROC) (2021), [10.26024/p6mv-en77](https://doi.org/10.26024/p6mv-en77).
- [35] J. P. Edelen and C. C. Hall, Autoencoder based analysis of rf parameters in the Fermilab low energy linac, *Information* **12**, 238 (2021).
- [36] M. Schwabacher and K. Goebel, A survey of artificial intelligence for prognostics, in *Proceedings of Artificial Intelligence for Prognostics: Papers from the AAAI Fall Symposium* (AAAI Press, Menlo Park, CA, 2007), pp. 107–114.